Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of the Claims:

1. (Currently Amended) A computer-implemented method for increasing the performance of a virtual machine that uses an interpreter to execute virtual machine program instructions, the computer-implemented method comprising:

obtaining, by an interpreter, a virtual machine program instruction to be executed by the virtual machine;

determining, by the interpreter, whether the virtual machine program instruction is a branch instruction;

determining, by the interpreter, whether a basic block is present in a code cache that stores native code corresponding to virtual machine program instructions when it is determined that the program instruction is a branch instruction, wherein the basic block is associated with a case block of the interpreter associated with the virtual machine program instruction and the basic block includes ~~including~~ native code that ~~represents the program instruction~~ has been previously interpreted and executed by the interpreter, the code cache being associated with the virtual machine; [[and]]

executing, by the interpreter, the code included in the basic block from said code cache when it is determined that the basic block is present in the code cache and the program instruction is a branch instruction;

interpreting, by the interpreter, the virtual machine instruction when said determining determines that the basic block associated with the case block of the interpreter is not present in said code cache, wherein said interpreting generates native code for the virtual machine instruction; and

copying the native code generated by the interpreter into said cache code after the interpreting of the code by the interpreter when said determining determines that the basic block associated with the case block of the interpreter is not present in said code cache.

2. (Cancelled)

3. (Original) A computer-implemented method as recited in claim 2 further including:

allocating space in the code cache for the code corresponding to the program instruction; and

providing the code corresponding to the program instruction with a label.

4. (Original) A computer-implemented method as recited in claim 3 further including placing the label in a table of labels.

5. (Previously Presented) A computer-implemented method as recited in claim 2 wherein determining whether the basic block is present in the code cache includes searching through a table of labels to determine if a target associated with the program instruction has a matching label in the table of labels.

6. (Original) A computer-implemented method as recited in claim 2 wherein the program instruction is a bytecode, and wherein the bytecode is executed by an interpreter of the virtual machine.

7. (Original) A computer-implemented method as recited in claim 2 wherein the code cache is a native code cache, and the code corresponding to the program instruction is native code.

8. (Original) A computer-implemented method as recited in claim 1 wherein the program instruction is a bytecode and the code cache is a native code cache.

9. (Previously Presented) A computer-implemented method as recited in claim 8 further including interpreting the bytecode when the determining determines that the program instruction is not a branch instruction.

10. (Previously Presented) A computer-implemented method as recited in claim 1 further including:

computing a target using the program instruction, wherein the determining of whether the basic block is present in the code cache includes determining if the code cache includes any basic blocks which correspond to the target.

11- 27 (Cancelled)

28. (New) A computer system for increasing the performance of a virtual machine that uses an interpreter to execute virtual machine program instructions, wherein the computer system is capable of operating to:

obtain a virtual machine program instruction to be executed by the virtual machine;

determine whether the virtual machine program instruction is a branch instruction;

determine whether a basic block is present in a code cache that stores native code corresponding to virtual machine program instructions when it is determined that the program instruction is a branch instruction, wherein the basic block is associated with a case block of the interpreter associated with the virtual machine program instruction and the basic block includes native code that has been previously interpreted and executed by the interpreter;

execute the code included in the basic block from said code cache when it is determined that the basic block is present in the code cache and the program instruction is a branch instruction;

interpret the virtual machine instruction when said determining determines that the basic block associated with the case block of the interpreter is not present in said code cache, wherein said interpreting generates native code for the virtual machine instruction; and

copy the native code generated by the interpreter into the cache code after the interpreting of the code by the interpreter when said determining determines that the basic block associated with the case block of the interpreter is not present in said code cache.

29. (New) A computer readable medium including computer program code for increasing the performance of a virtual machine that uses an interpreter to execute virtual machine program instructions, wherein computer readable medium comprises:

computer program code for obtaining by an interpreter a virtual machine program instruction to be executed by the virtual machine;

computer program code for determining by the interpreter whether the virtual machine program instruction is a branch instruction;

computer program code for determining by the interpreter whether a basic block is present in a code cache that stores native code corresponding to virtual machine program instructions when it is determined that the program instruction is a branch instruction, wherein the basic block is associated with a case block of the interpreter associated with the virtual machine program instruction and the basic block includes native code that has been previously interpreted and executed by the interpreter, the code cache being associated with the virtual machine;

computer program code for executing by the interpreter the code included in the basic block from said code cache when it is determined that the basic block is present in the code cache and the program instruction is a branch instruction;

computer program code for interpreting by the interpreter the virtual machine instruction when said determining determines that the basic block associated with the case block of the interpreter is not present in said code cache, wherein said interpreting generates native code for the virtual machine instruction; and

computer program code for copying the native code generated by the interpreter into the cache code after the interpreting of the code by the interpreter when said determining determines that the basic block associated with the case block of the interpreter is not present in said code cache.